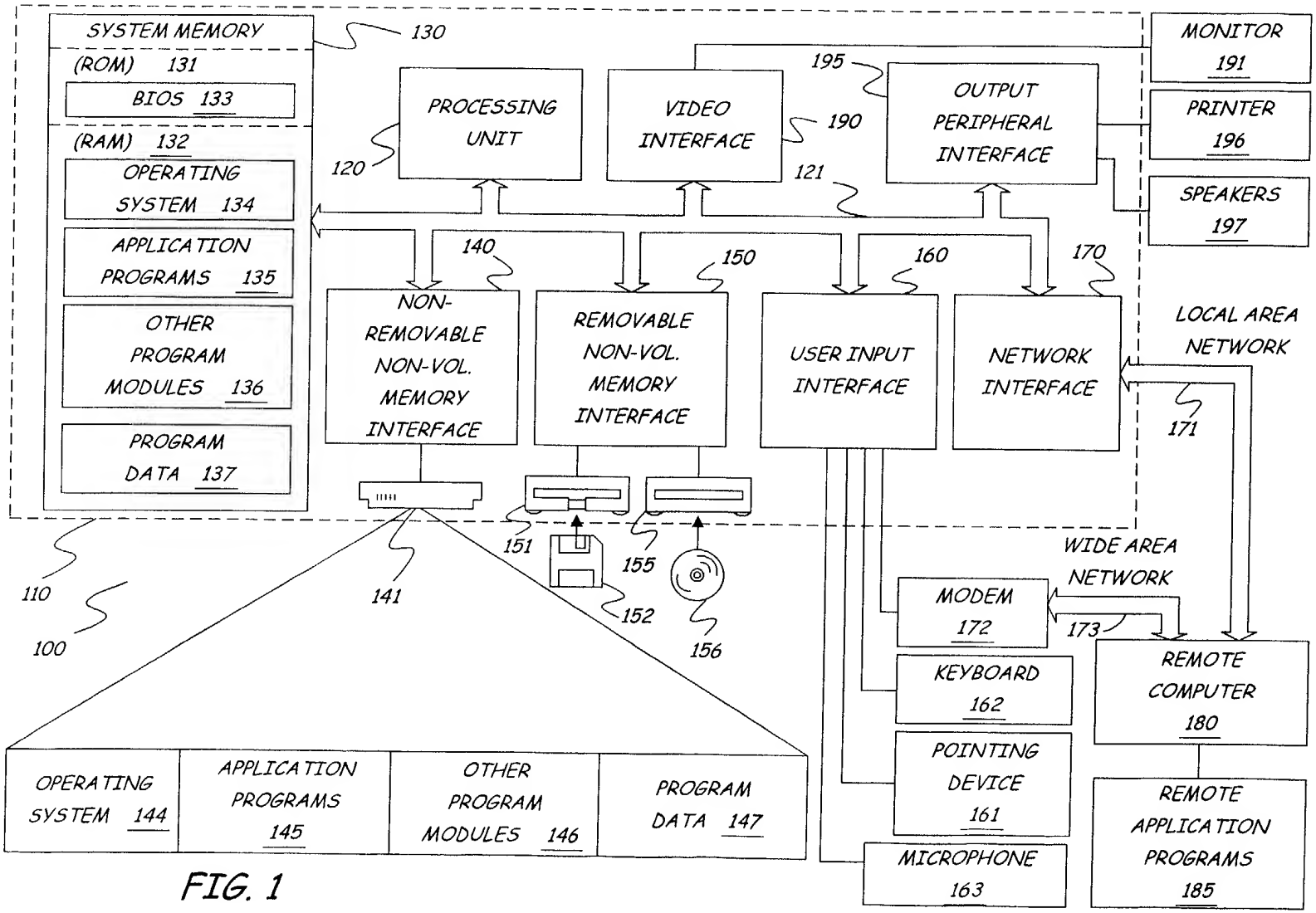

SYSTEM MEMORY
130
(ROM) 131
BIOS 133
(RAM) 132
OPERATING SYSTEM 134
APPLICATION PROGRAMS 135
OTHER PROGRAM MODULES 136
PROGRAM DATA 137
PROCESSING UNIT
120
VIDEO INTERFACE
190
195
OUTPUT PERIPHERAL INTERFACE
121
MONITOR 191
PRINTER 196
SPEAKERS 197
140
150
160
170
NON-REMOVABLE NON-VOL. MEMORY INTERFACE
REMOVABLE NON-VOL. MEMORY INTERFACE
USER INPUT INTERFACE
NETWORK INTERFACE
LOCAL AREA NETWORK
171
110
100
141
151
155
152
156
WIDE AREA NETWORK
MODEM 172
173
KEYBOARD 162
POINTING DEVICE 161
MICROPHONE 163
REMOTE COMPUTER 180
REMOTE APPLICATION PROGRAMS 185
OPERATING SYSTEM 144
APPLICATION PROGRAMS 145
OTHER PROGRAM MODULES 146
PROGRAM DATA 147

FIG. 1

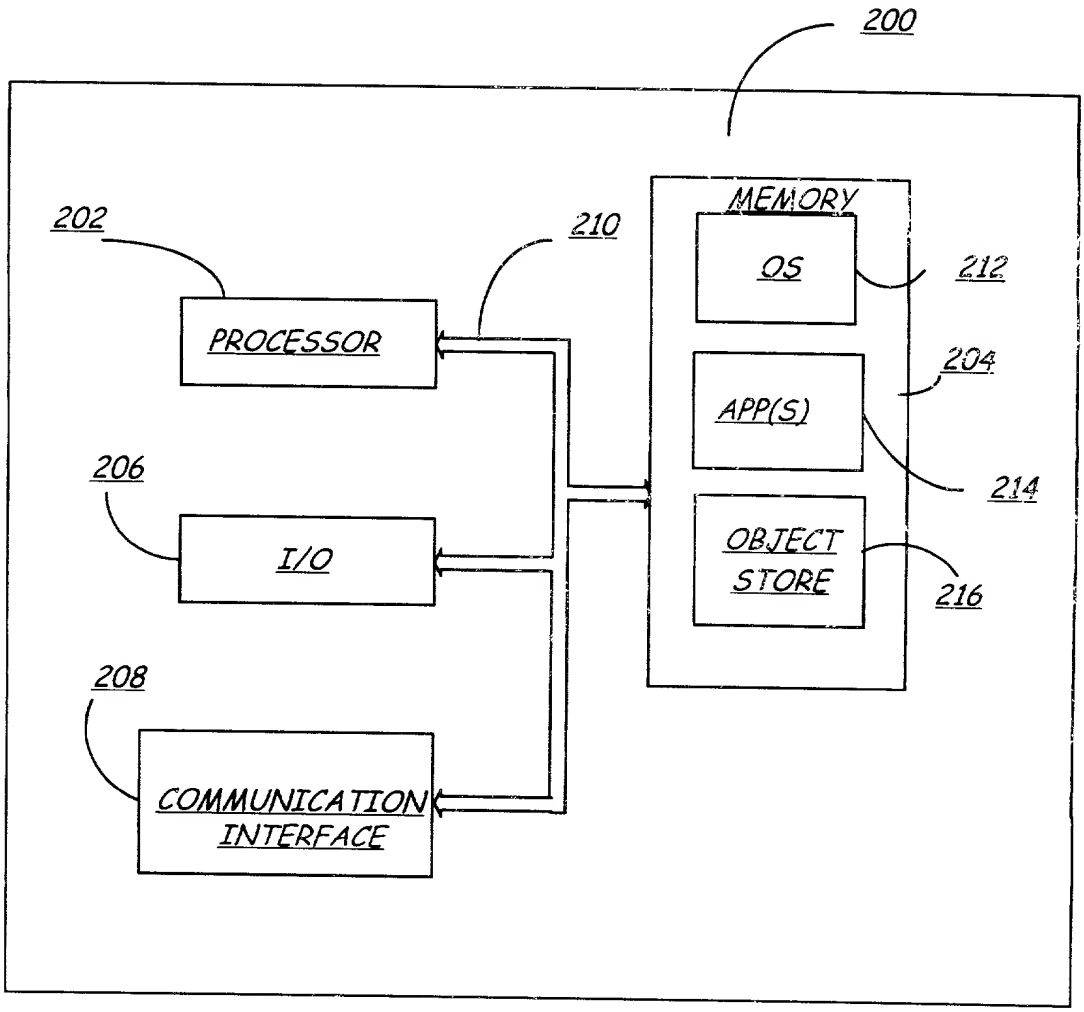

200
MEMORY
202
210
OS
212
PROCESSOR
204
APP(S)
214
206
I/O
OBJECT STORE
216
208
COMMUNICATION INTERFACE

FIG. 2

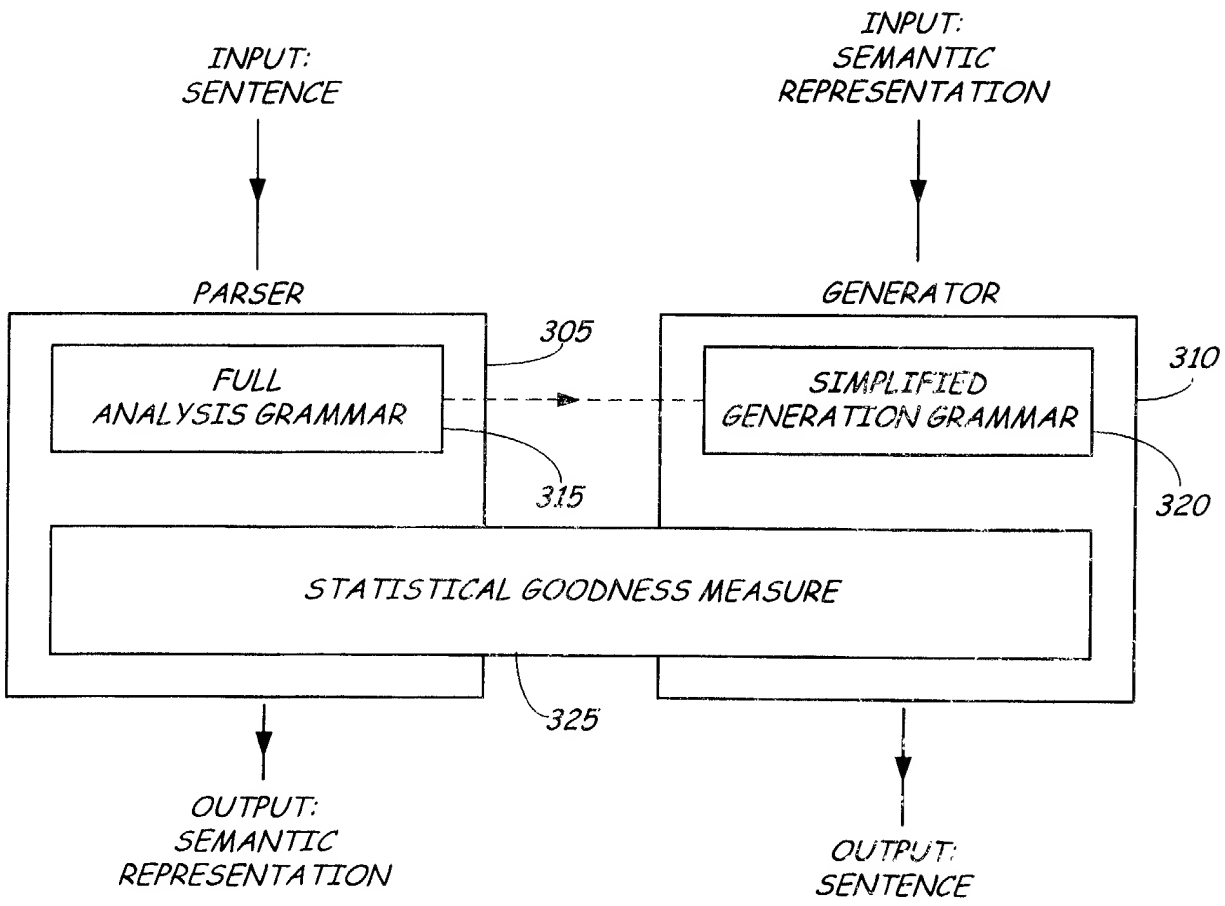

INPUT:
SENTENCE
INPUT:
SEMANTIC
REPRESENTATION
PARSER
GENERATOR
305
310
FULL
ANALYSIS GRAMMAR
SIMPLIFIED
GENERATION GRAMMAR
315
320
STATISTICAL GOODNESS MEASURE
325
OUTPUT:
SEMANTIC
REPRESENTATION
OUTPUT:
SENTENCE

FIG. 3

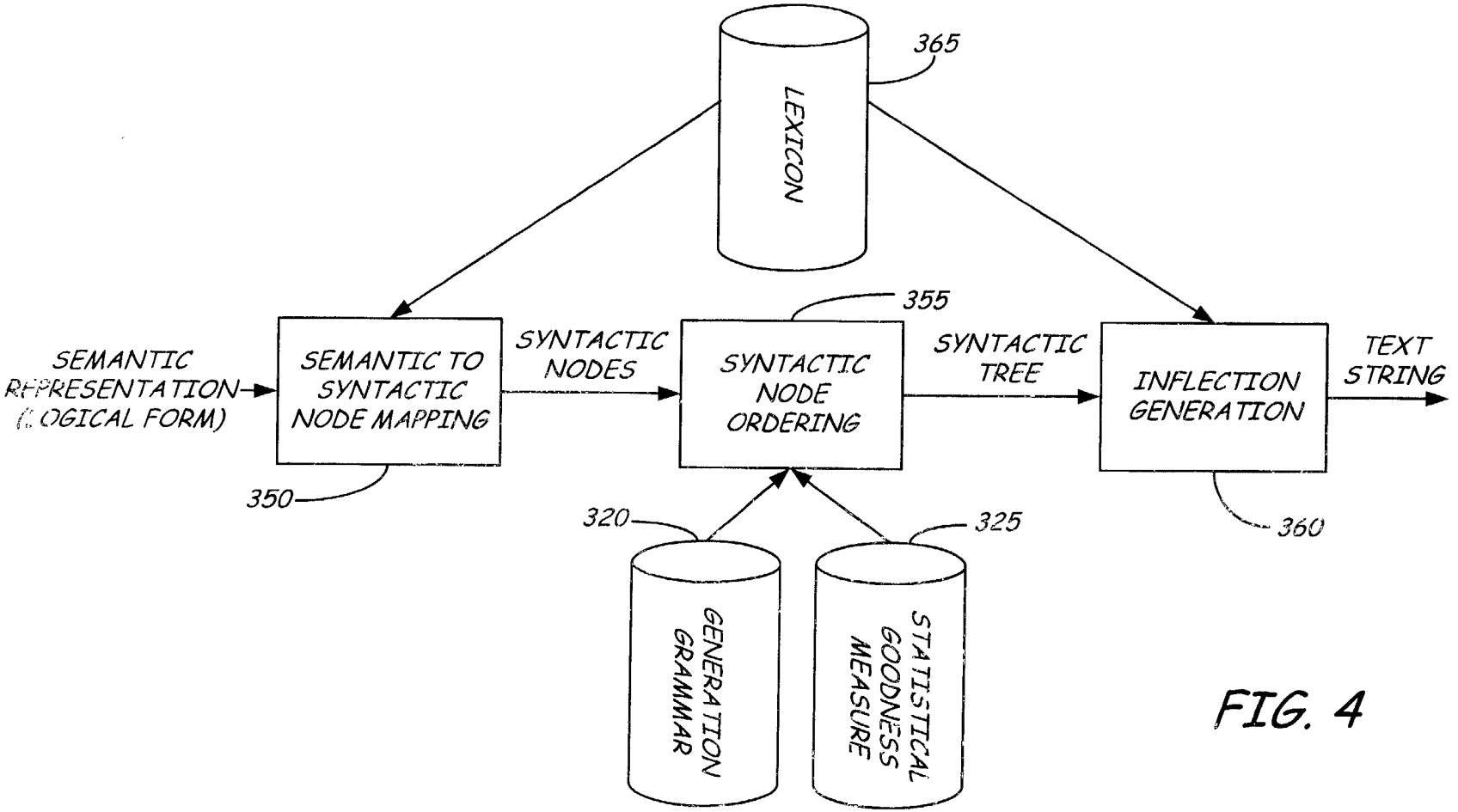

LEXICON
365
SEMANTIC REPRESENTATION (LOGICAL FORM)
SEMANTIC TO SYNTACTIC NODE MAPPING
350
SYNTACTIC NODES
355
SYNTACTIC NODE ORDERING
SYNTACTIC TREE
INFLECTION GENERATION
360
TEXT STRING
320
GENERATION GRAMMAR
325
STATISTICAL GOODNESS MEASURE

FIG. 4

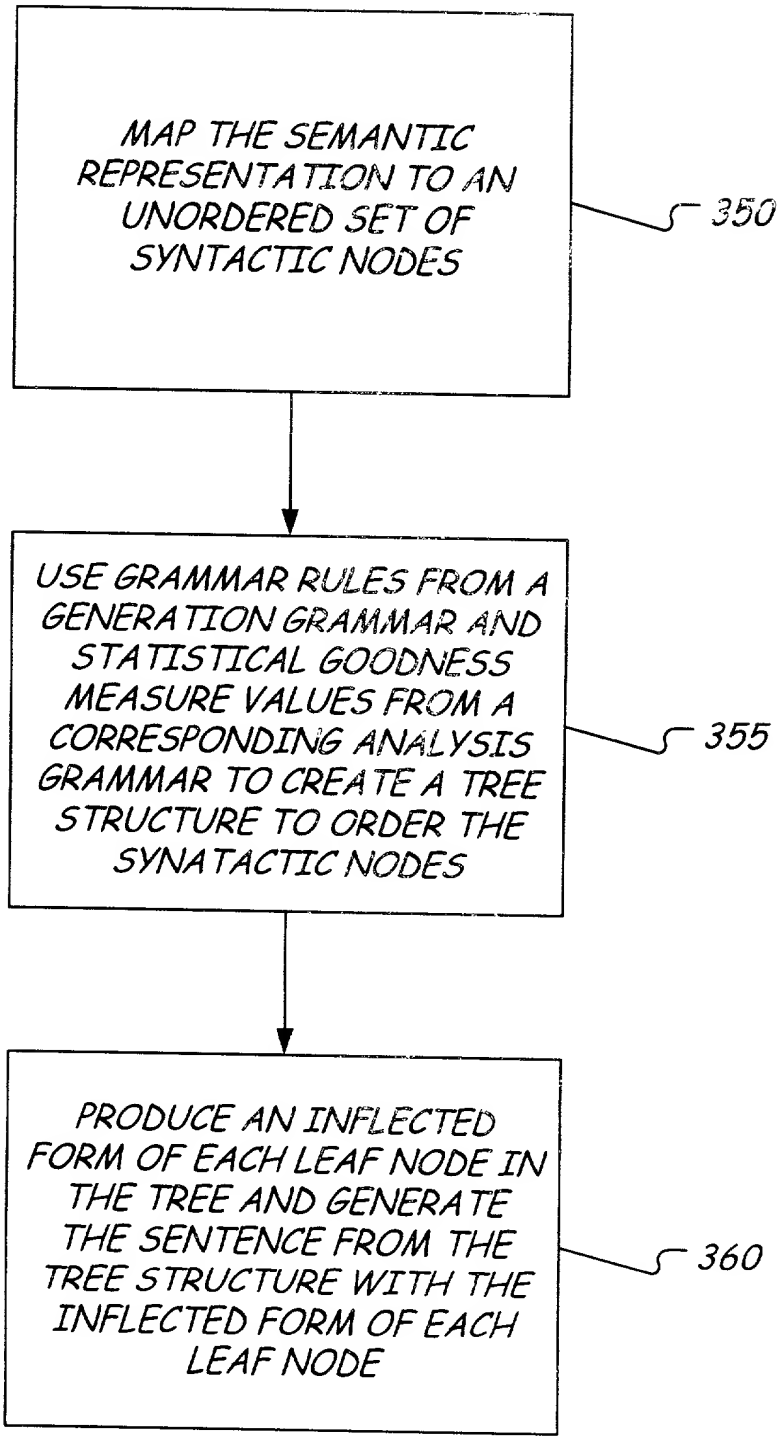

MAP THE SEMANTIC REPRESENTATION TO AN UNORDERED SET OF SYNTACTIC NODES
350
USE GRAMMAR RULES FROM A GENERATION GRAMMAR AND STATISTICAL GOODNESS MEASURE VALUES FROM A CORRESPONDING ANALYSIS GRAMMAR TO CREATE A TREE STRUCTURE TO ORDER THE SYNATACTIC NODES
355
PRODUCE AN INFLECTED FORM OF EACH LEAF NODE IN THE TREE AND GENERATE THE SENTENCE FROM THE TREE STRUCTURE WITH THE INFLECTED FORM OF EACH LEAF NODE
360

FIG. 5

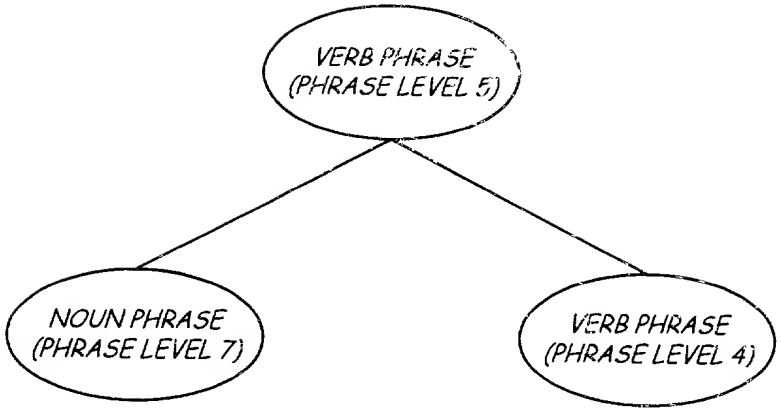


FIG. 6

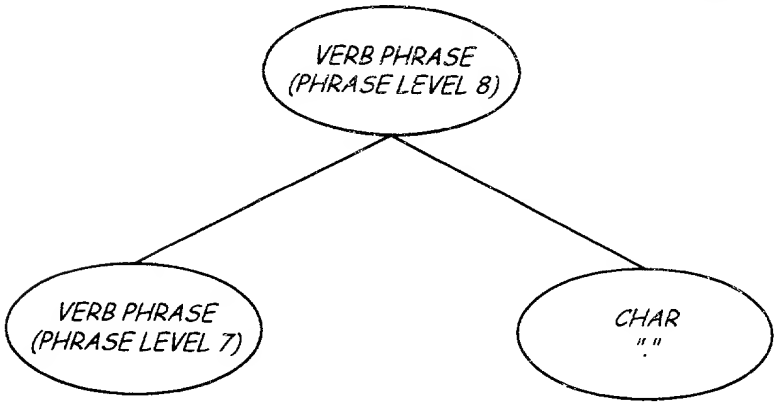


FIG. 8

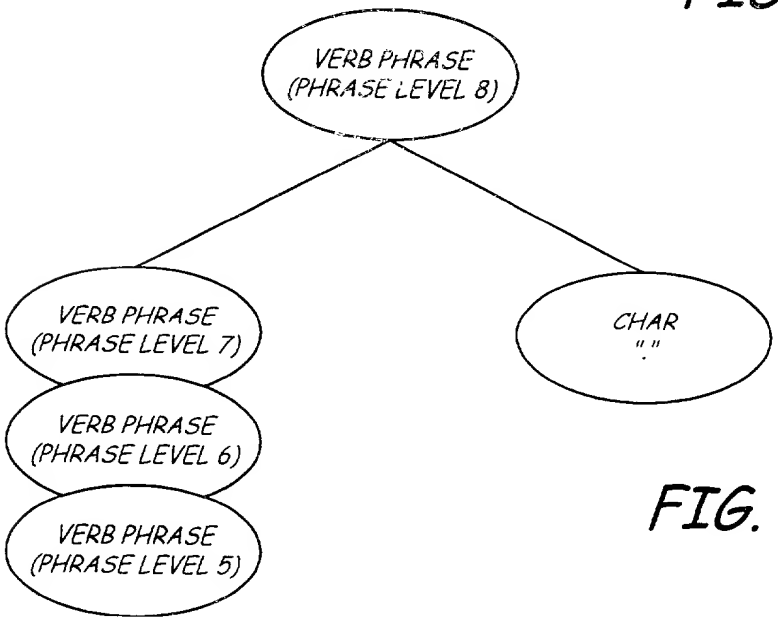


FIG. 9

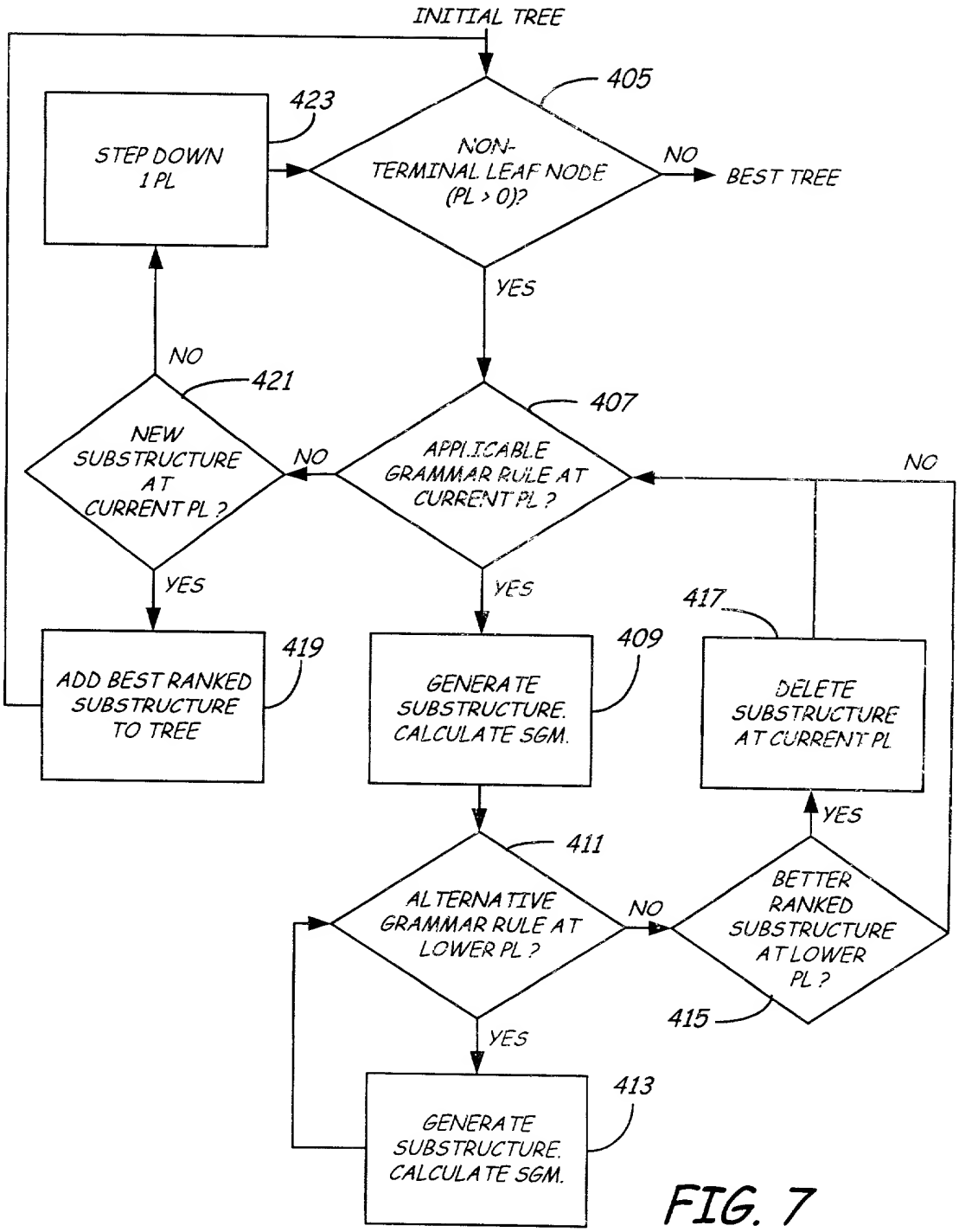

INITIAL TREE
405
NON-TERMINAL LEAF NODE (PL > 0)?
NO
BEST TREE
YES
423
STEP DOWN 1 PL
NO
421
NEW SUBSTRUCTURE AT CURRENT PL ?
NO
407
APPLICABLE GRAMMAR RULE AT CURRENT PL ?
NO
YES
YES
419
ADD BEST RANKED SUBSTRUCTURE TO TREE
409
GENERATE SUBSTRUCTURE. CALCULATE SGM.
417
DELETE SUBSTRUCTURE AT CURRENT PL
YES
411
ALTERNATIVE GRAMMAR RULE AT LOWER PL ?
NO
BETTER RANKED SUBSTRUCTURE AT LOWER PL ?
415
YES
413
GENERATE SUBSTRUCTURE. CALCULATE SGM.

FIG. 7

09695356 - 021498

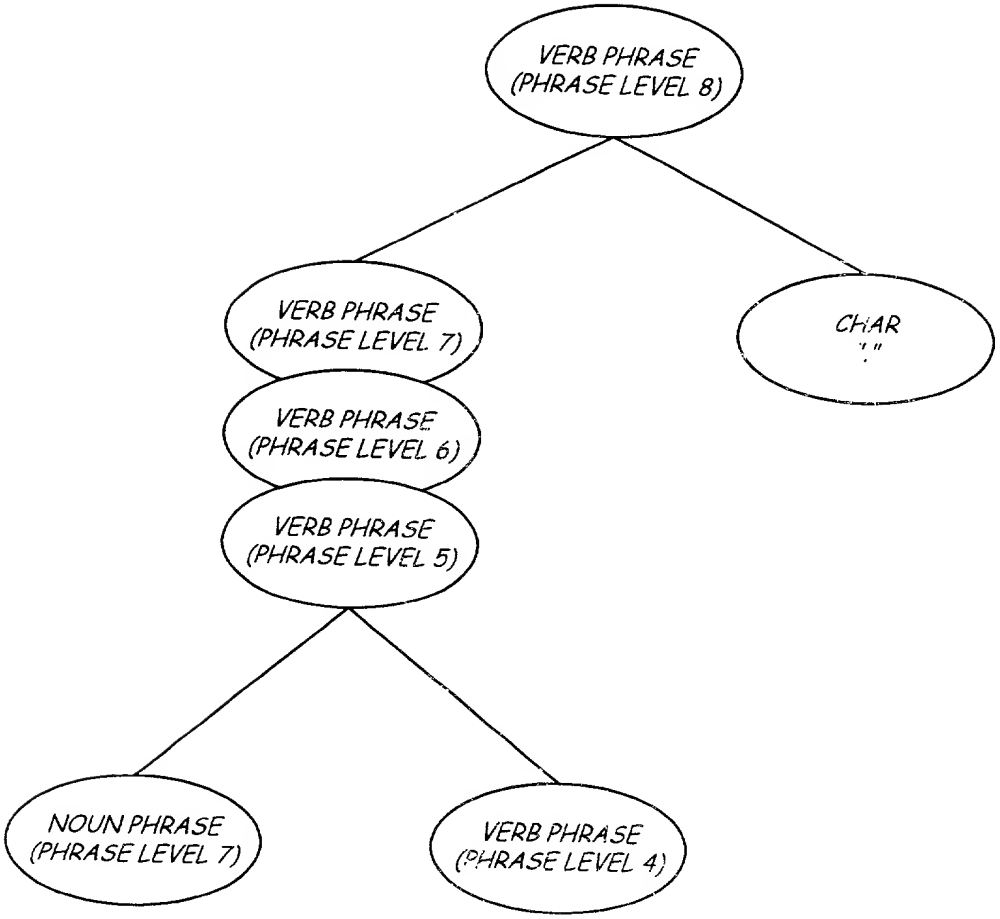


FIG. 10

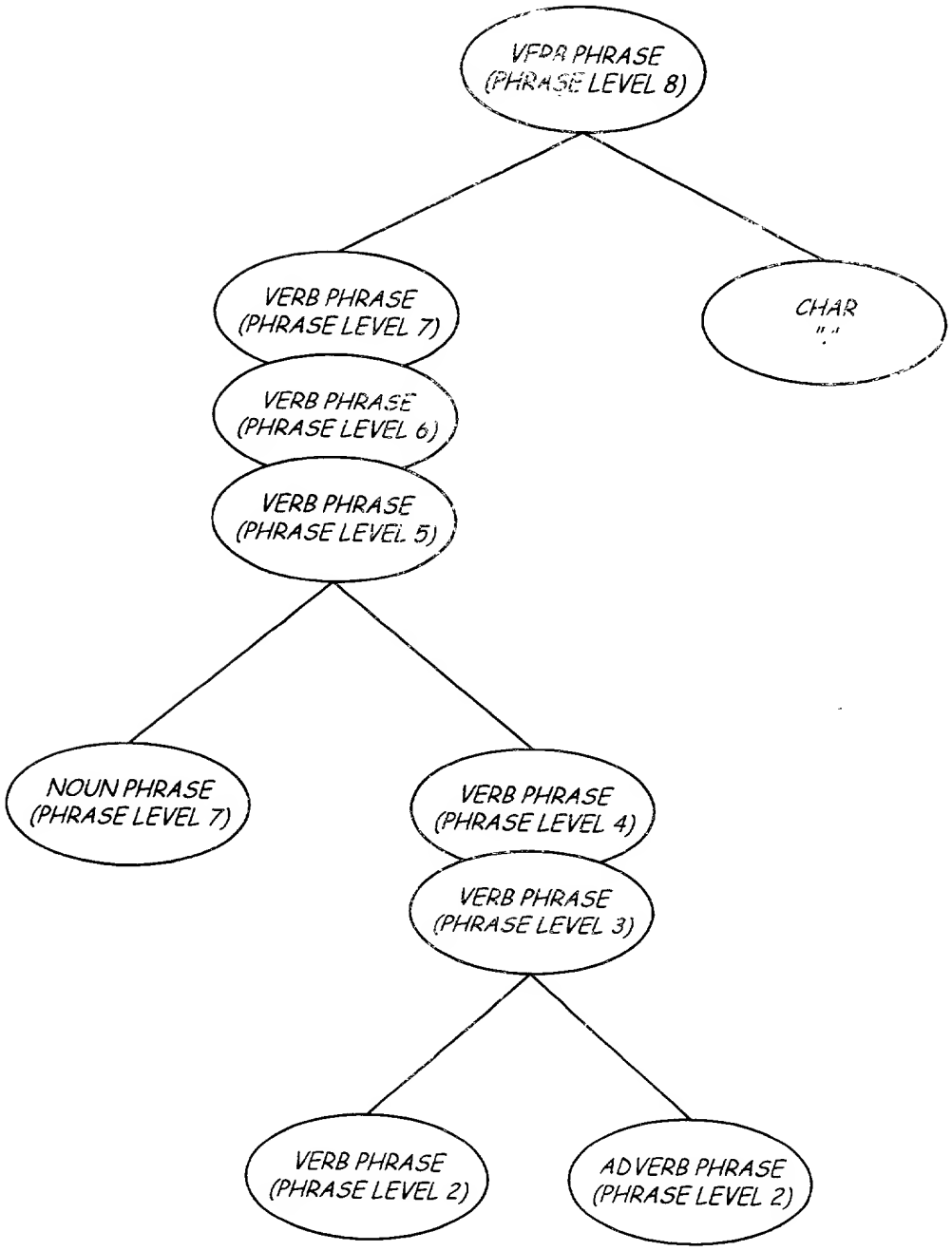
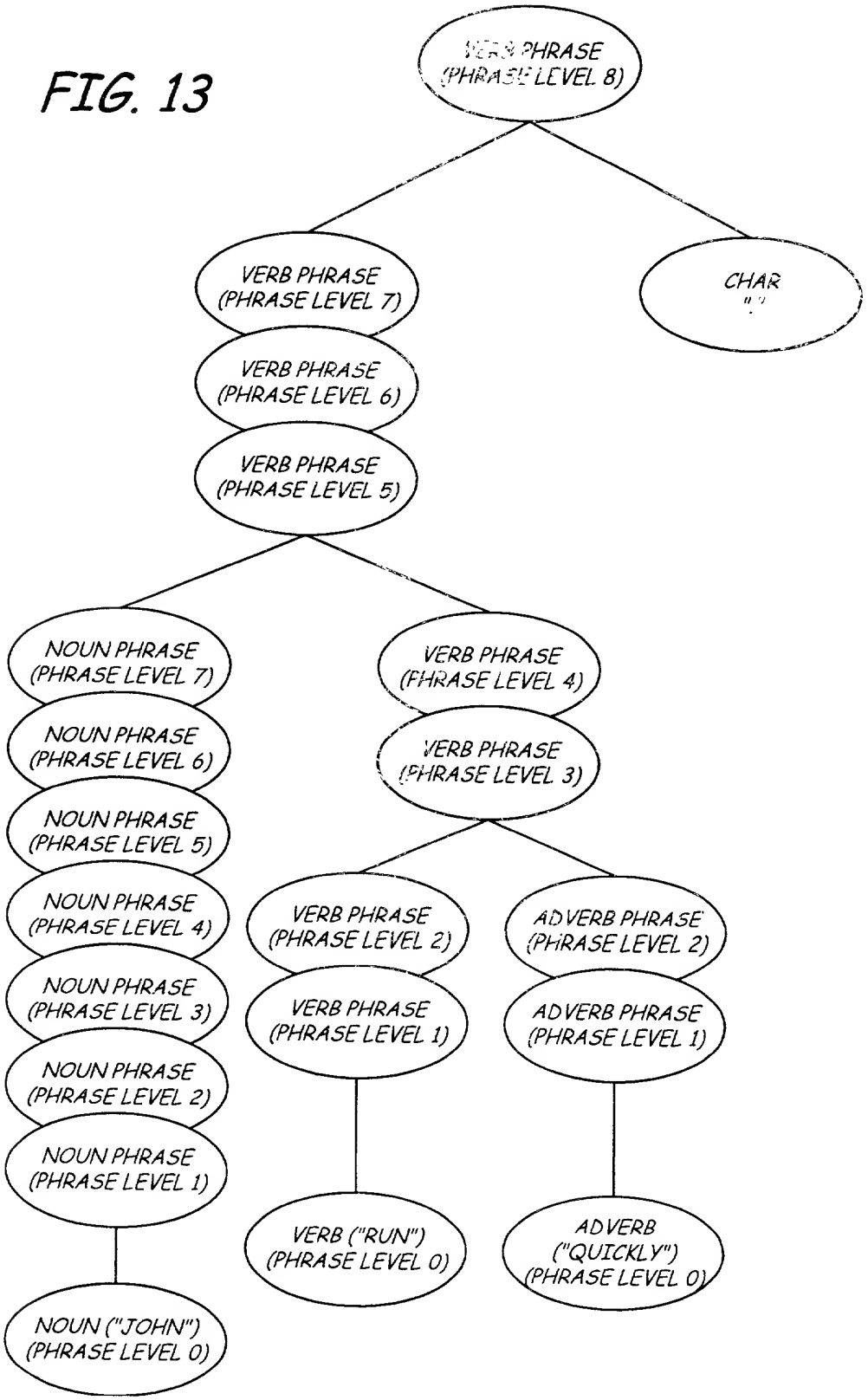

VERB PHRASE
(PHRASE LEVEL 8)
VERB PHRASE
(PHRASE LEVEL 7)
CHAR
VERB PHRASE
(PHRASE LEVEL 6)
VERB PHRASE
(PHRASE LEVEL 5)
NOUN PHRASE
(PHRASE LEVEL 7)
VERB PHRASE
(PHRASE LEVEL 4)
VERB PHRASE
(PHRASE LEVEL 3)
VERB PHRASE
(PHRASE LEVEL 2)
ADVERB PHRASE
(PHRASE LEVEL 2)

FIG. 11

VERB PHRASE
(PHRASE LEVEL 8)
VERB PHRASE
(PHRASE LEVEL 7)
CHAR
" "
VERB PHRASE
(PHRASE LEVEL 6)
VERB PHRASE
(PHRASE LEVEL 5)
NOUN PHRASE
(PHRASE LEVEL 7)
VERB PHRASE
(PHRASE LEVEL 4)
VERB PHRASE
(PHRASE LEVEL 3)
VERB PHRASE
(PHRASE LEVEL 2)
ADVERB PHRASE
(PHRASE LEVEL 2)
VERB PHRASE
(PHRASE LEVEL 1)
VERB ("RUN")
(PHRASE LEVEL 0)

FIG. 12

FIG. 13



09/966,536 - 012402

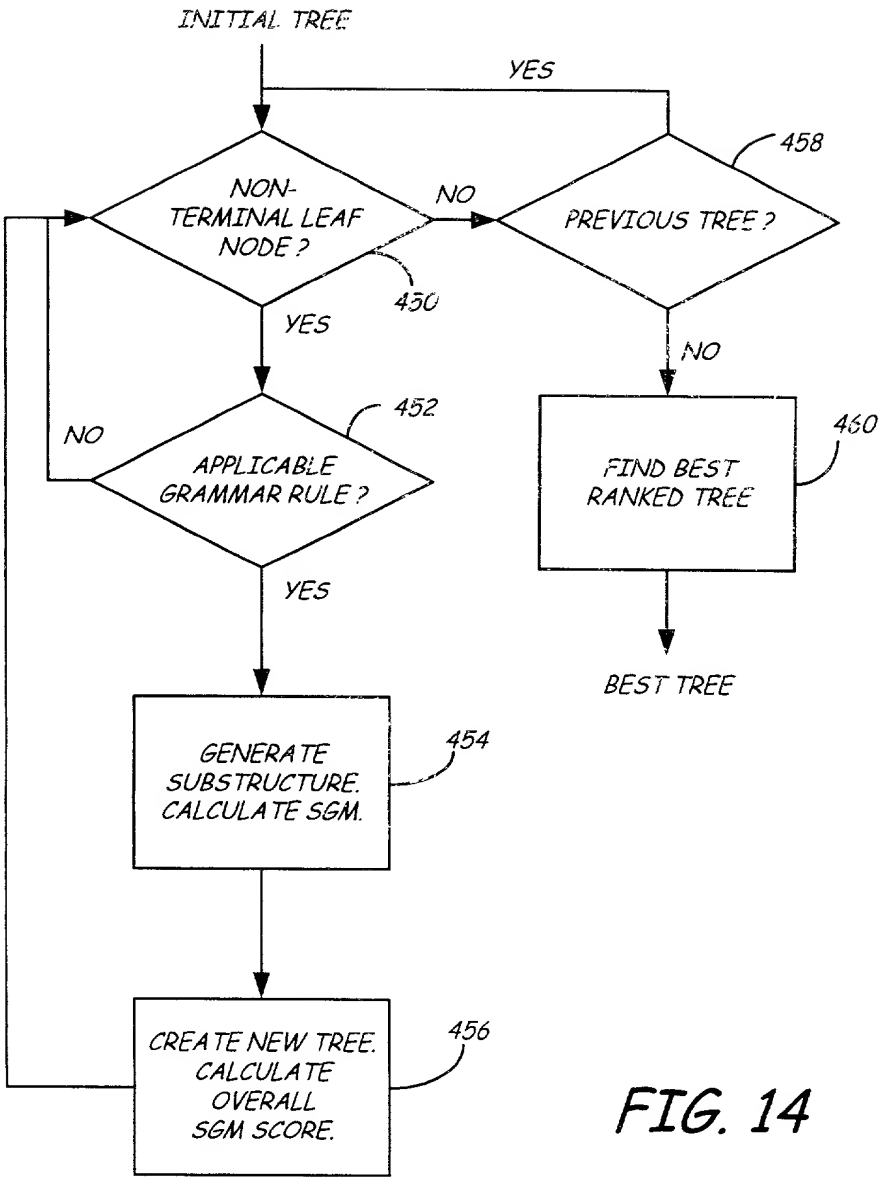

INITIAL TREE
YES
NON-TERMINAL LEAF NODE ?
NO
PREVIOUS TREE ?
458
430
YES
NO
NO
APPLICABLE GRAMMAR RULE ?
452
460
FIND BEST RANKED TREE
YES
BEST TREE
GENERATE SUBSTRUCTURE. CALCULATE SGM.
454
CREATE NEW TREE. CALCULATE OVERALL SGM SCORE.
456

FIG. 14

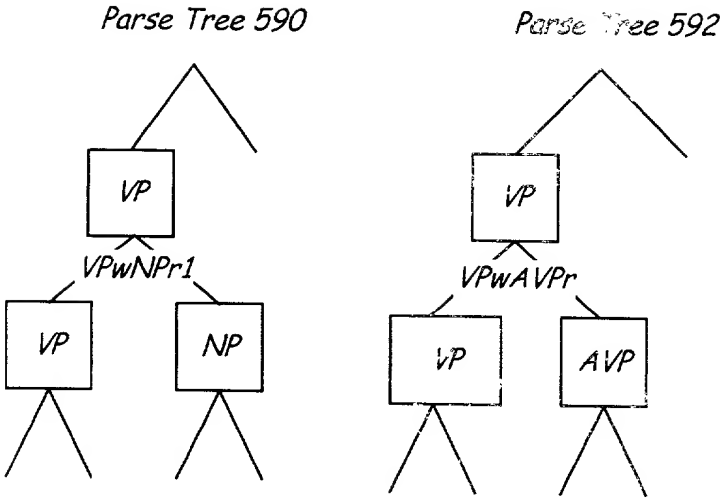


FIG. 15A

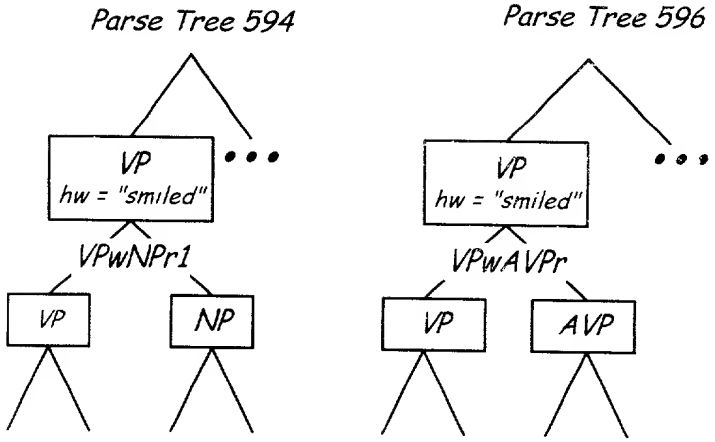


FIG. 15B

Rule Name	Rule				Description (and an example)
VERBtoVP	VP	→	VERB		Promote a verb into a verb phrase
PredAdj	VP	→	VP	AJP	Add on an adjective. *"It has been found effective"*
VPwNPrl	VP	→	VP	NP	Add direct object to a VP *"I hit Bill"*
Perfect	VP	→	VP	VP	Adds the VP "have" to a VP *"Joe had gone"*
SubjPQuant	VP	→	NP	VP	Add quantifiers to the VP like "all", "both", ... *"The children have all gone"*
VPwNPl	VP	→	NP	VP	Add noun subject to a VP *"John jumped."*
SubjectAJP	VP	→	AJP	VP	Add AJP subject to a VP *"More surprising is his attitude"*
InvertAJPwS	VP	→	AJP	VP	Add final modifier, AJP, to the left of a VP *"More remarkable still, he went"*
Topicalization	VP	→	NP	VP	Add object of VP, NP, to the left of a VP *"A better-looking man I have not seen"*

FIG. 16

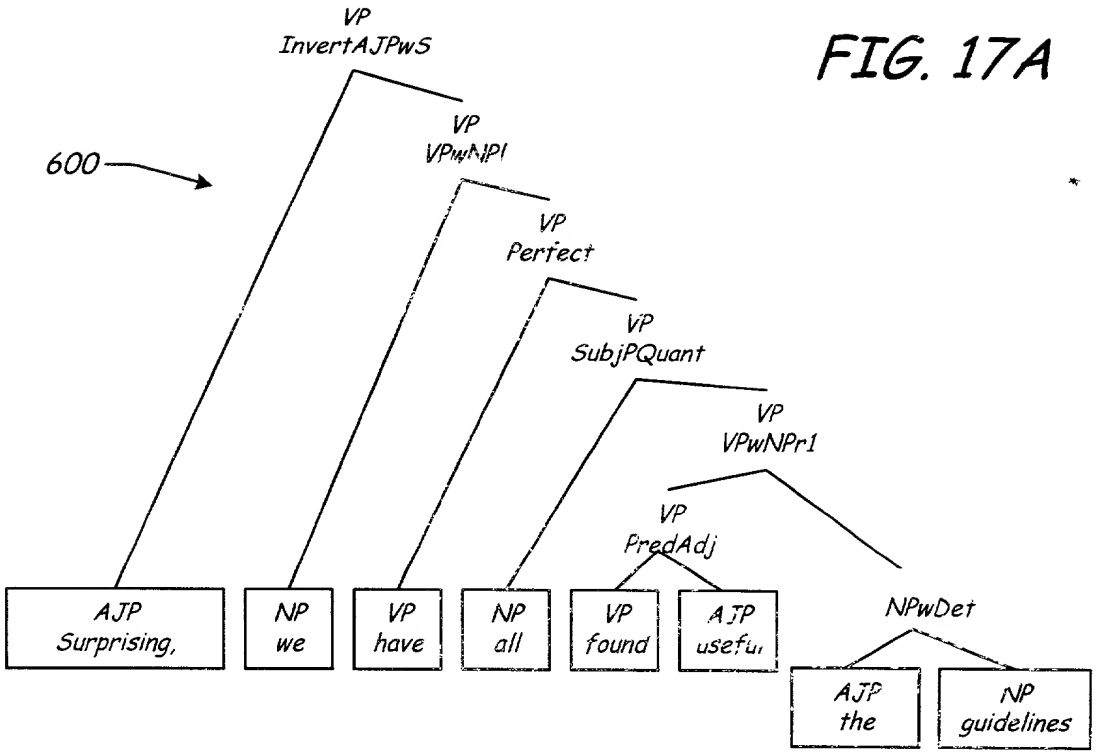


FIG. 17A

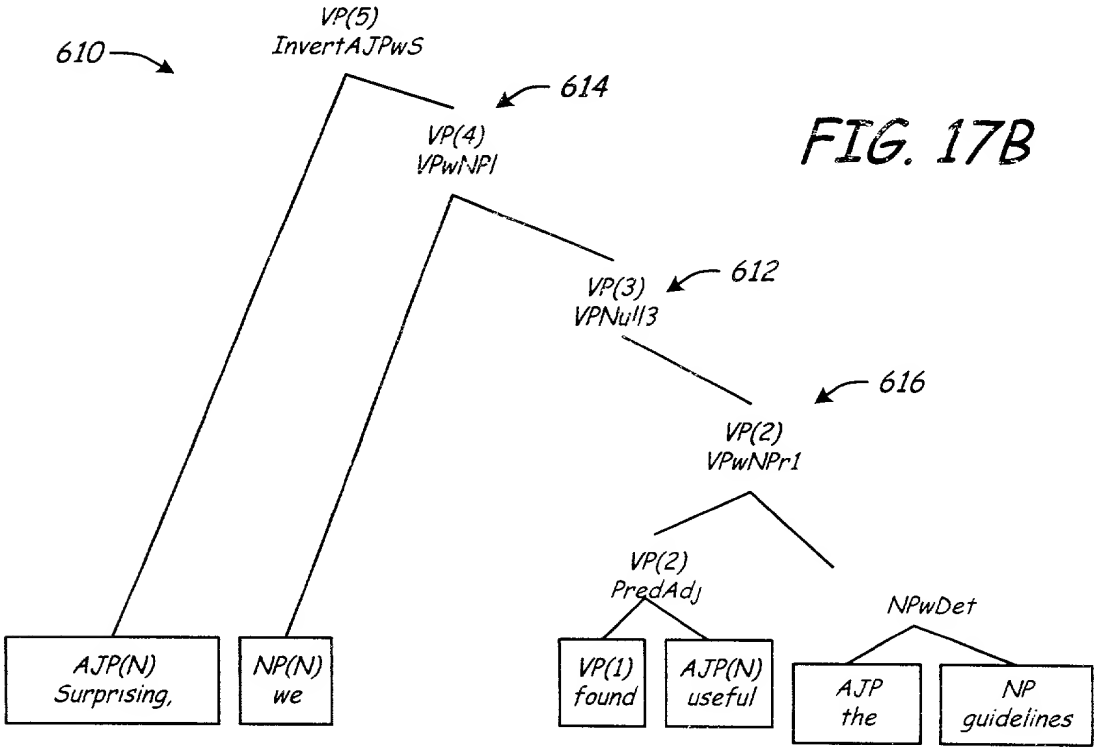


FIG. 17B

Attorney Docket No. 60655-0160

20nZTD" 02560560

Phrase Level	Rule Name	Rule			Level Description
1	VERBtoVP	VP(1) →	VERB(PL_Verb_Max)		Create a VP
2	PredAdj	VP(2) →	VP(1)	AJP(PL_AJP_Max)	Add post modifiers
	VPwNPr1	VP(2) →	VP(1)	NP(PL_AJP_Max)	
	VPNull2	VP(2) →	VP(1)		
3	Perfect	VP(3) →	VP(1)	VP(2,3)	Add "have" and quantifiers
	SubjPQuant	VP(3) →	NP(PL_NP_Max)	VP(2,3)	
	VPNull3	VP(3) →	VP(2)		
4	VPwNPl	VP(4) →	NP(PL_NP_Max)	VP(3)	Add subject
	SubjectAJP	VP(4) →	AJP(PL_AJP_Max)	VP(3)	
	VPNull4	VP(4) →	VP(3)		
5	InvertAJPwS	VP(5) →	AJP(PL_AJP_Max)	VP(4)	Add modifiers to VPs that have a subject
	Topicalization	VP(5) →	NP(PL_NP_Max)	VP(4)	
	VPNull5	VP(5) →	VP(4)		

FIG. 18

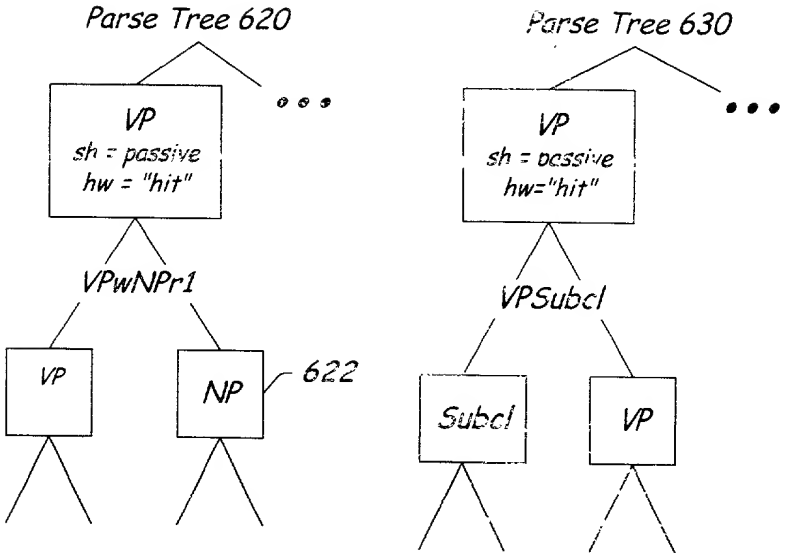


FIG. 19

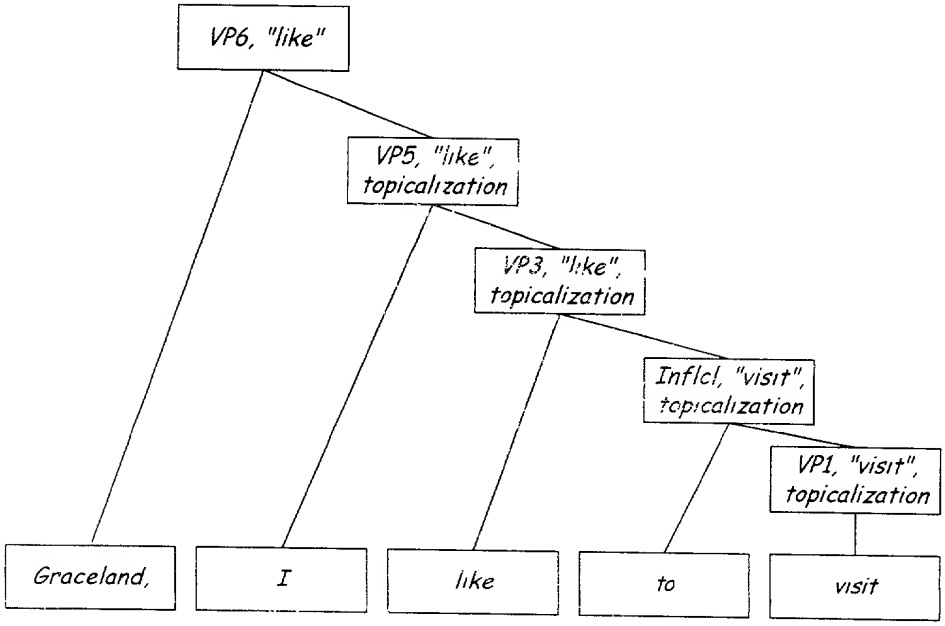


FIG. 20

09595546 - 071400

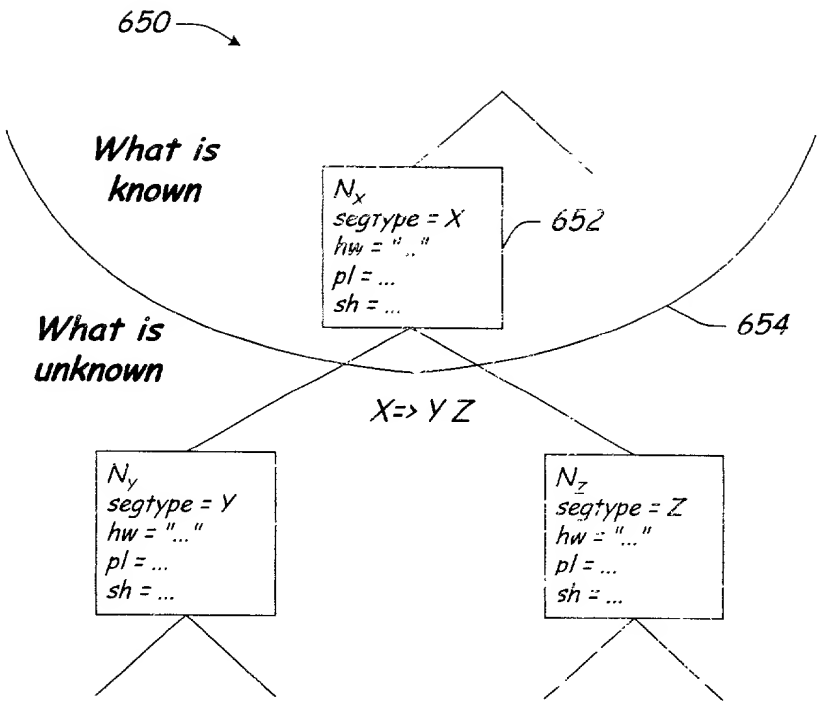

650
What is known
What is unknown
N_X
segtype = X
hw = "..."
pl = ...
sh = ...
652
654
X=> Y Z
N_Y
segtype = Y
hw = "..."
pl = ...
sh = ...
N_Z
segtype = Z
hw = "..."
pl = ...
sh = ...

FIG. 21